



PATENT
Atty. Docket No. 678-760(P9995)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT(S): Jae-Yoel KIM, et al.
SERIAL NO.: 10/054,400
FILED: October 22, 2001
FOR: APPARATUS AND METHOD FOR GENERATING BLOCK
CODE IN A MOBILE COMMUNICATION SYSTEM
DATED: April 22, 2002

Assistant Commissioner
for Patents
Washington, D.C. 20231

TRANSMITTAL OF PRIORITY DOCUMENT

Sir:

Enclosed is a certified copy of Korean Patent Appln. No.

2000/62084 filed on October 21, 2000 and from which priority is claimed under
35 U.S.C. §119.

Respectfully submitted,

Paul J. Farrell
Reg. No. 33,494
Attorney for Applicant(s)

DILWORTH & BARRESE, LLP
333 Earle Ovington Blvd.
Uniondale, NY 11553
(516) 228-8484

CERTIFICATION UNDER 37 C.F.R. 1.8(a)

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.D. 20231

Dated: April 22, 2002

Barbara Evers

COPY OF PAPERS
ORIGINALLY FILED



#4



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원번호 : 특허출원 2000년 제 62084 호
Application Number PATENT-2000-0062084

출원년월일 : 2000년 10월 21일
Date of Application OCT 21, 2000

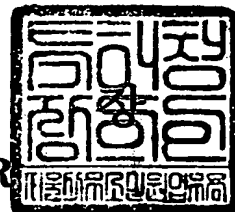
출원인 : 삼성전자 주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2001 년 10 월 25 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】 특허출원서
【권리구분】 특허
【수신처】 특허청장
【참조번호】 0001
【제출일자】 2000. 10. 21
【국제특허분류】 H04M
【발명의 명칭】 통신시스템에서 블록부호 발생 장치 및 방법
【발명의 영문명칭】 APPARATUS FOR GENERATING BLOCK CODE AND METHOD THEREOF IN COMMUNICATION SYSTEM

【출원인】

【명칭】 삼성전자 주식회사
【출원인코드】 1-1998-104271-3

【대리인】

【성명】 이건주
【대리인코드】 9-1998-000339-8
【포괄위임등록번호】 1999-006038-0

【발명자】

【성명의 국문표기】 김재열
【성명의 영문표기】 KIM, Jae Yoel
【주민등록번호】 700219-1047637
【우편번호】 435-042
【주소】 경기도 군포시 산본2동 산본9단지 백두아파트 960동 1401호

【국적】 KR

【발명자】

【성명의 국문표기】 최호규
【성명의 영문표기】 CHOI, Ho Kyu
【주민등록번호】 681204-1787524
【우편번호】 463-500
【주소】 경기도 성남시 분당구 구미동 무지개아파트 1204-303호
【국적】 KR



1020000062084

출력 일자: 2001/10/26

【취지】

특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인
이건주 (인)

【수수료】

【기본출원료】

20 면 29,000 원

【가산출원료】

26 면 26,000 원

【우선권주장료】

0 건 0 원

【심사청구료】

0 항 0 원

【합계】

55,000 원

【첨부서류】

1. 요약서·명세서(도면)_1통

**【요약서】****【요약】**

본 발명에 따른, k 개의 정보비트의 입력으로 n 개의 부호심볼을 출력하는 블록 부호 생성장치가, 상기 k 개의 정보비트 입력으로 상기 n 개보다 많은 부호심볼을 출력하는 확장된 리드물러(Extended Reed Muller) 부호기와, 상기 부호기로부터 출력되는 상기 부호심볼들 중 미리 주어지는 천공패턴에 의해 소정의 개수를 천공하여 상기 n 개의 부호심볼을 출력하는 천공기를 포함한다.

【대표도】

도 4

【색인어】

퀄리티 매칭 지시자(Quality Matching Indicator), 블록부호(Block code), 오류정정부호, 부호기, 복호기

【명세서】

【발명의 명칭】

통신시스템에서 블록부호 발생 장치 및 방법{APPARATUS FOR GENERATING BLOCK CODE AND METHOD THEREOF IN COMMUNICATION SYSTEM}

【도면의 간단한 설명】

도 1은 퀄리티 매칭 지시자(Quality Matching Indicator)의 사용 예를 도시하는 도면

도 2는 프레임을 전송하는 송신기의 구조를 도시하는 도면

도 3은 도 3은 수신기의 구성을 도시하는 도면

도 4는 (24,7)부호기의 구성을 나타내는 도면

도 5는 본 발명에 따른 선형 오류정정부호의 개략적인 부호화 구조를 나타내는 도면

도 6은 본 발명에 따른 확장된 리드플러 부호기의 마스크함수의 생성과정을 나타내는 도면

도 7a은 본 발명의 제 1실시예에 따른 선형 오류정정부호의 부호화 구조를 나타내는 도면

도 7b은 본 발명의 제 2실시예에 따른 선형 오류정정부호의 부호화 구조를 나타내는 도면

도 8은 본 발명의 제 1실시예에 따른 부호기의 동작에 대한 플로우 차트를 나타내는 도면

도 9는 본 발명에 따른 선형 오류정정부호의 복호화 구조를 나타내는 도면

도 10은 본 발명에 따른 복호기에서 비교기의 동작에 대한 플로우 차트를 나타내는 도면

도 11은 본 발명의 제 2실시예에 따른 부호기의 동작에 대한 플로우 차트를 나타내는 도면

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

- <13> 본 발명은 통신시스템에서 블록부호(Block code) 발생장치 및 방법에 관한 것으로, 특히 블록부호를 이용해 퀄리티 매칭 지시자(Quality Matching Indicator)를 부호화하기 위한 장치 및 방법에 관한 것이다.
- <14> 일반적으로 부호분할을 수행하는 이동통신 시스템(이하 IMT2000 시스템이라 칭한다)에서는 하나의 물리적 채널안에 음성서비스, 화상서비스, 데이터서비스와 같은 여러 가지 서비스의 프레임을 같이 전송한다. 상기와 같은 여러 가지 서비스의 프레임은 한 프레임에서 병합되어져서 전송되어지는데, 상기와 같이 병합되어져 전송되어질 때, 각각의 서비스의 Quality를 보장하기 위해서 각각의 서비스 프레임 데이터의 비율이 달라질 수 있다.따라서, 상기와 같이 각각의 서비스 프

레이미 데이터의 비율이 달라질 때, 송신기는 상기 비율에 대한 정보를 수신기측에 알려주어야 한다.

<15> 따라서, 상기와 같이 각각의 서비스 프레임 데이터의 비율에 대한 정보를 알려주는 역할을 하는 것이 퀄리티 매칭 지시자(Quality Matching Indicator)이고, 상기 퀄리티 매칭 지시자(Quality Matching Indicator)는 올바르게 수신될 수 있어야 한다. 상기 퀄리티 매칭 지시자(Quality Matching Indicator)의 사용 예는 첨부된 도면 도 1과 같다.

<16> 상기 도 1은 본 발명의 실시예에 따른 패킷 데이터 서비스를 위한 순방향 링크 데이터 트래픽 MAC 채널 구조를 보여주는 도면이다. 상기 패킷 데이터 서비스를 위한 물리 계층의 최소 전송 단위는 1,536 칩(chip)으로 구성되는 슬롯(slot)이며, 이는 1.25msec의 지속시간을 갖는다. 상기 도 1b를 참조하면, 데이터 트래픽 MAC 채널(Data Traffic MAC Channel: DTMACCH)은 제1채널(I-ch)과 제2채널(Q-ch)로 구성된다. 상기 제1채널 I-ch는 QoS(또는 퀄리티) 매칭 지시채널(Matching Indication Channel: QMICH)로 사용된다. 상기 제2 채널 Q-ch는 왈시 공간 지시 부채널(Walsh Space Indication Subchannel: WSISCH)과, 역방향 활성 지시 부채널(Reverse Activity Indication Subchannel: RAISCH)로 사용된다. 1슬롯 동안에 WSISCH와 RAISCH는 각각 1,280칩과 256칩 구간을 차지하며, 이 채널들은 멀티플렉싱되어 DTMACCH의 직각위상채널을 구성한다.

<17> 도 2는 상기 퀄리티 매칭 지시자를 포함하는 프레임을 전송하는 송신기의 구조를 도시한다. 도 2를 참조하여 송신구조를 설명하면, 먼저 한 슬롯에 전송되어질 7비트의 퀄리티 매칭 지시자 비트가 (24,7)부호기 200에 입력되면, 상기 부호기

200은 상기 퀄리티 매칭 지시자 비트를 부호화하여 24심볼의 0과 1의 값을 가지는 부호화 심볼을 출력한다. 그러면, 신호변환기 210은 입력된 심볼이 0은 1로, 1은 -1로 변환하여 1과 -1의 값을 가지는 심볼들을 출력한다. 그러면, 상기 신호 변환기 210에서 출력된 24개의 부호화 심볼들은 승산기 220으로 심볼단위로 입력되며, 각각의 한 심볼씩 입력됨과 동시에 길이 64인 월시부호인 월시 커버(Walsh Cover)가 승산기220에 입력되어 칩단위로 승산되어져, 한심볼이 입력될때마다 64개의 칩이 출력되어 총 $24 \times 64 = 1536$ 칩이 출력되어져 전송되어진다.

<18> 도 3은 상기 도 2의 송신기 구조에 대응하는 수신기의 간략한 구성을 도시하고 있다. 도 3를 참조하여 수신구조를 설명하면, 먼저 수신된 신호가 심볼누적기 320에 입력되어지면, 상기 심볼누적기 320은 상기 수신된 신호와 상기 송신기에서 사용한 월시커버를 승산하여 역확산(despeading)하고, 이를 심볼단위로 누적하여 출력한다. 그러면, 상기 심볼누적기 320으로부터 출력되어진 심볼들은 상기 송신기의 부호기에 대응하는 복호기 300에 입력되어져, 복호화되어진 후, 복호화된 퀄리티 매칭 지시자를 출력하게 된다.

<19> 상기 퀄리티 매칭 지시자의 값은 7비트로 표현되어진다. 상기 상기 퀄리티 매칭 지시자의 값은 수신단에서 각 서비스들의 프레임을 해석하기 위해 반드시 필요한 정보이므로, 전송 오류가 발생한다면 수신단에서 각 서비스들의 프레임을 올바르게 해석하지 못하는 경우가 발생한다. 따라서 상기 상기 퀄리티 매칭 지시자의 값은 전송중에 오류가 발생한다 할지라도 수신단에서 상기 퀄리티 매칭 지시자의 전송중에 발생하는 오류를 정정할 수 있도록 오류정정부호를 사용하여 부호화된다.

<20> 일반적으로 이진 선형 부호(Binary Linear Codes)의 오류 정정 능력은 이진 선형 부호의 각 부호간 최단 거리에 따라 결정되는데 최적부호(optimal code)가 되기 위한 이진 선형 부호의 입력과 출력값에 따른 부호간의 최단 거리는 하기 [1]과 문헌에 상세히 설명되어 있다.

<21> 참조문헌[1] An Updated Table of Minimum-Distance Bounds for Binary Linear Codes

<22> (A.E. Brouwer and Tom Verhoeff, IEEE Transactions on information Theory, VOL 39, NO. 2, MARCH 1993)

<23> 상기 도 2에서 전송되는 퀄리티 매칭 지시자의 값이 7비트이고, 부호화되는 값이 24비트임을 생각하면 참조문헌 1에서 기술되어 있는 바와 같이, 요구되는 최적 부호의 각 부호간의 최단거리는 10이다. 만일, 후술되는 도 5의 오류정정방식 부호화에서 성능이 좋지 않는다면 동일한 채널환경에서 퀄리티 매칭지시자의 오류 확률이 커진다. 따라서, 상기 퀄리티 매칭 지시자의 오류가 발생하여 데이터프레임의 각각의 서비스프레임의 비율을 잘못 판단하고 각각의 데이터프레임을 복호화하면 데이터프레임의 오류율을 증가시키게 된다. 따라서, 퀄리티 매칭 지시자를 부호화하는 오류정정부호기는 오류율을 최소화하는 것이 중요하다.

【발명이 이루고자 하는 기술적 과제】

<24> 따라서 본 발명의 목적은 (24,7) 블록부호 발생장치 및 방법을 제공함에 있다.

- <25> 본 발명의 다른 목적은 퀄리티 메칭 지시자(Quality Matching Indicator)를 부호화하기 위한 장치 및 방법을 제공함에 있다.
- <26> 본 발명의 또 다른 목적은 (32,7) 블록부호를 이용해 (24,7) 블록부호를 생성하기 위한 장치 및 방법을 제공함에 있다.
- <27> 상기 목적들을 달성하기 위한, k개의 정보비트의 입력으로 n개의 부호심볼을 출력하는 블록 부호 생성장치가, 상기 k개의 정보비트 입력으로 상기 n개보다 많은 부호심볼을 출력하는 확장된 리드물러(Extended Reed Muller) 부호기와, 상기 부호기로부터 출력되는 상기 부호심볼들 중 미리 주어지는 천공패턴에 의해 소정의 개수를 천공하여 상기 n개의 부호심볼을 출력하는 천공기를 포함하는 것을 특징으로 한다.

【발명의 구성 및 작용】

- <28> 이하 본 발명의 바람직한 실시예의 상세한 설명이 첨부된 도면들을 참조하여 설명될 것이다. 도면들 중 참조번호들 및 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 참조번호들 및 부호들로 나타내고 있음에 유의해야 한다. 하기에서 본 발명을 설명함에 있어, 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다.
- <29> 본 발명은 상기한 바와 같이 퀄리티 메칭 지시자(Quality Matching Indicator) 비트를 사용하는 CDMA 시스템에서 퀄리티 메칭 지시자(Quality

Matching Indicator)를 부호화하는 방법에 있어서 최적부호를 생성할 수 있는 방법으로 확장된 리드물러(Extended Reed Muller)부호를 천공한 부호기를 CDMA시스템에 적용하는 하는 것이다.

<30> 선형 오류정정부호(Linear Error Correcting Code)의 성능을 나타내는 척도(measure)로서는 오류정정부호의 부호어(codeword)의 해밍 거리(Hamming distance)의 분포가 있는데, 이는 각각의 부호어에서 0이 아닌 심볼의 개수를 의미한다. 즉, 0111이 어떤 부호어라면 이 부호어에 포함된 1의 개수, 즉, 해밍거리는 3이다. 이 때, 여러 부호어의 해밍거리 값들 중 가장 작은 값을 최소거리를 d_{min} (minimum distance)라고 칭한다. 선형 오류정정부호(Linear Error Correcting Code)에 있어서 상기의 최소거리가 클수록 오류정정성능이 우수하다.[참조문헌:'The Theory of Error-Correcting Codes' - F.J.Macwilliams, N.J.A. Sloane, North-Holland]

<31> 도 4는 본 발명의 실시 예에 따른 7비트로 표현되는 퀄리티 매칭 지시자를 오류정정 부호화하기 위한 부호기 장치를 도시하고 있다.

<32> 상기 도 4에 도시된 바와 같이, 1 - 128 사이의 퀄리티 매칭 지시자값은 7비트로 표현되어, (32,7) 확장된 리드물러 (Extended Reed Muller) 부호기 400으로 입력되고, 상기 부호기 400은 상기 퀄리티 매칭 지시자 비트를 부호화하여 32심볼의 퀄리티 매칭 지시자 부호어를 출력된다. 그러면 상기 32심볼의 부호 심볼들은 천공기410에 입력되고, 천공기 410은 입력된 부호 심볼들중 8개의 부호심볼들을 천공함으로써 총 24개의 부호심볼들을 출력한다. 상기 도 4에서 입력정보비트는 퀄리

터 매칭 지시자값의 7비트 표현형식으로, 7비트가 되지 않는 지시자 값은 MSB(Most significant bit, 제일 왼쪽의 bit)부터 0의 값으로 채워져 7비트 길이로 만들어 부호화한다.

<33> 상기의 확장된 리드물러(Extended Reed Muller)는 특정 시퀀스(sequence)와 m-시퀀스의 합으로 이루어지는 시퀀스로부터 유추해 낼 수 있는데, 상기의 시퀀스들의 합을 원소로하는 시퀀스군을 선형 오류정정부호로 사용하기 위해서는 시퀀스군의 최소거리가 커야한다. 이러한 시퀀스군으로는 카자미 시퀀스(Kasami sequence)군, 골드 시퀀스(Gold sequence)군 및 커독코드(Kerdock Code)군과 같은 시퀀스군들이 있다. 상기의 시퀀스들은 전체길이 $L=2^{2m}$ 일 때, 최소거리가 $(2^{2m} - 2^m)/2$ 이고, $L=2^{2m+1}$ 일 때, 최소거리가 $2^{2m} - 2^m$ 이다. 즉, 전체길이가 32일 때, 최소거리는 12이다.

<34> 본 발명에서는 상기와 같은 시퀀스군을 사용하여 우수한 성능을 가지는 선형 오류정정부호인 확장된 오류정정부호를 생성하는 생성장치 및 방법에 대해 설명하기로 한다.

<35> 코드이론(Coding Theory)에서, m-시퀀스를 월시부호로 만드는 열치환 함수는 존재한다. 이 때, 상기와 같은 특정 시퀀스(sequence)와 m-시퀀스의 합으로 이루어지는 시퀀스들을 상기의 열치환함수로 열치환 하게 되면, m-시퀀스 성분은 월시부호가 되고, 특정 시퀀스(sequence)부분은 상기의 월시부호와의 합의 최소거리가 상기의 성질을 만족하게 되는데, 이를 이하 마스크함수라 칭하기로 한다.

도 5는

상기와 같은 선형 오류정정부호의 부호화 구조를 나타낸다. 한편, 상기 도 5는 상기 도 4의 부호기 400에 대응하는 구조를 나타낸다. 상기 도 5를 참조하면, 입력되는 인덱스 정보는 상기 퀄리티 매칭 지시자 7비트에 대응한다. 상기 7비트에서 상위 2비트는 마스크 발생기 502에 입력되는 인덱스 정보로 사용되고, 상기 7비트에서 하위 5비트는 월시부호 발생기 504에 입력되는 인덱스 정보로 사용된다. 상기 마스크 발생기 502는 입력되는 2비트 인덱스 정보에 대응하는 길이 32인 마스크 신호를 발생한다. 상기 월시부호 발생기 504는 입력되는 5비트 인덱스 정보에 대응하는 길이 32인 월시부호를 발생한다. 가산기 510은 상기 마스크 발생기 502 및 상기 월시부호 발생기 504의 출력을 가산하여 길이 32인 부호심볼을 출력한다.

<36> 이하 상기 골드 시퀀스군을 사용하여, $(2^n, n+k)$ 코드(단, $k=1, \dots, n+1$)를 생성하는 경우 상기와 같은 마스크함수의 생성방법을 설명한다. 실제로, 골드 시퀀스는 서로 다른 특정의 m -시퀀스의 합으로 표현되는 것은 널리 알려진 사실이다. 따라서, 상기의 $(2^n, n+k)$ 코드를 생성하기 위해서는, 먼저, 길이 2^n-1 인 골드 시퀀스를 생성하여야 하는데, 생성다항식 $f_1(x)$ 과 $f_2(x)$ 으로 생성되어지는 두 m -시퀀스의 합은 골드시퀀스가 된다. 또한, 각각의 m -시퀀스 $m(t)$ 는 생성다항식이 정해지면 <수학식1>과 같이 트레이스 함수(Trace function)를 이용하여 구할 수 있다.

<37> <수학식 1>

<38> $m_1(t) = \text{Tr}(A \alpha^t), t=0, 1, \dots, 30$

- <39> 단, $Tr(a) = \sum_{k=0}^{n-1} a^{2^k}$, $a \in GF(2^n)$ 이다.
- <40> 상기 <수학식1>에서 A는 m-시퀀스의 초기치에 따라 결정되는 값이다.
- <41> 도 6은 상기의 시퀀스군중 골드 시퀀스군을 사용하여, $(2^n, n+k)$ 코드(즉, $n+k$ 비트의 정보비트가 입력되면 2^n 비트의 부호화심볼이 출력되는)를 생성하는 경우 상기한 마스크함수의 생성과정을 나타낸다. 골드 시퀀스는 서로 다른 특정의 m-시퀀스의 합으로 표현되어진 다는 것은 널리 알려진 사실이다. 따라서, 상기의 $(2^n, n+k)$ 코드를 생성하기 위해서는, 먼저, 길이 2^n-1 인 골드 시퀀스를 생성하여야 하는데, 생성다항식 $f_1(x)$ 과 $f_2(x)$ 으로 생성되어지는 두 m-시퀀스의 합은 골드시퀀스가 된다.
- <42> 상기 도 6을 참조하면, 610 단계에서는 상기의 <수학식1>에 의해 생성다항식 $f_1(x)$ 로 생성되어지는 m-시퀀스 $m_1(t)$ 와 $f_2(x)$ 으로 생성되어지는 m-시퀀스 $m_2(t)$ 를 구한다. 그러면, 620단계에서는 상기의 m-시퀀스 $m_2(t)$ 를 가지고 하기의 <수학식 2>에서 나타난 월시부호로 만드는 열치환 함수 $\sigma(t)$ 를 구한다.
- <43> 【수학식 2】 $\sigma: \{0, 1, 2, \dots, 2^n - 2\} \rightarrow \{1, 2, \dots, 2^n - 1\}$
- <44>
$$\sigma(t) = \sum_{i=0}^{n-1} m_i(t) 2^{n-1-i}$$
- <45> 그러면, 630단계에서는 상기 m-시퀀스 $m_2(t)$ 를 0부터 30번까지 순회(cyclic shift)시켜 얻을 수 있는 7개의 시퀀스군을 상기의 $m_1(t)$ 을 월시부호로 만드는 열치환 함수 $\sigma(t)$ 의 역함수를 이용한 $\sigma^{-1}(t) + 2$ 로 열치환한 후, 각각의 시퀀스 맨 앞부분에 0을 덧붙임으로써 길이 2^n 인 2^n-1 개의 시퀀스군 $d_i(t)$,

$i=0, \dots, 2^n-1, t=1, \dots, 2^n$, 을 생성한다. 상기와 같이 630단계에서 생성되어지는 시퀀스군은 <수학식 3>과 같이 수식으로 표현할 수 있다.

<46> 【수학식 3】 $\{d_i(t) | t=1, \dots, 2^n, i=0, \dots, 2^n-2\}$

$$d_i(t) = \begin{pmatrix} 0, & \text{if } t=1 \\ m_d(t+i-2), & \text{if } t=1, 2, 3, \dots, 2^n \end{pmatrix}$$

<48> 상기와 같이 구해진 $d_i(t)$ 들은 상기 마스크 함수들로 31개의 마스크로 사용할 수 있다.

<49> 상기 구해진 $d_i(t)$ 들의 성질 중 한가지 성질은 상기의 마스크들중 두 개의 서로 다른 마스크들을 더하면 2^n-1 개마스크중 다른 하나의 마스크가 된다. 더 일반화시켜서 전부 0인 마스크를 포함하여 상기의 2^n-1 개의 마스크들은 2^n-1 개의 마스크중 특정한 n 개의 마스크의 임의의 합으로 모두 표현되어질 수 있다.

<50> 상기 $(2^n, n+k)$ 코드를 생성할 때, 총 필요로하는 부호어(Code word)의 개수는 모든 경우의 입력 정보비트의 가지수인 2^{n+k} 개이다. 이 때, $(2^n, n+k)$ 코드를 생성하기 위해 필요한 마스크의 개수는 $(2^{n+k}/2^n)-1 = 2^k-1$ 개이다. 또한, 이 때, 2^k-1 개의 마스크는 상기와 유사한 성질에 의해 k 개의 마스크의 임의의 합으로 모두 표현되어질 수 있다. 따라서, 상기의 k 개의 마스크를 고르는 방법이 필요하다.

<51> 상기의 k 개의 마스크를 선택하는 방법을 설명하면, 상기 도 6의 630단계에서 $m_1(t)$ 를 0부터 2^n-1 번까지 순회(cyclic shift)시켜 시퀀스군을 생성하는데, 이 때, $m_1(t)$ 를 i 번 순회(cyclic shift)시킨 m -시퀀스는 상기 <수학식1>을 사용하여 표현하면 $Tr(\alpha' \cdot \alpha')$ 가 된다. 즉, $m_1(t)$ 를 0부터 30번까지 순회(cyclic

shift)시켜 시퀀스군은 초기치 A 가 $1, \alpha, \dots, \alpha^{2^n-2}$ 에 따라서 생성되어지는 시퀀스들이다. 이 때, 갈로아체의 원소 $1, \alpha, \dots, \alpha^{2^n-2}$ 중, 선형독립인 k 개의 원소를 찾아서 상기 k 개의 원소를 초기치로 하는 시퀀스들을 선택하여 상기 도 6의 결과값들을 찾아 선택하면 상기와 같이 k 개의 마스크의 임의의 합으로 2^k-1 개 마스크 모두를 생성할 수 있다. 상기의 과정중 선형독립의 조건을 수학적식으로 나타내면 하기의 <수학식4>과 같다.

<52> 【수학식 4】 $\alpha_1, \dots, \alpha_{k-1}$: 선형 독립

<53>
$$\Leftrightarrow c_1\alpha_1 + c_2\alpha_2 + \dots + c_{k-1}\alpha_{k-1} \neq 0, \quad \forall c_1, c_2, \dots, c_{k-1}$$

<54> 상기의 일반화된 마스크 함수생성방법을 구체적인 예를 들어 설명하기 위해서 하기에서는 골드 시퀀스군을 사용하여, (32,7)코드를 생성하는 경우 상기 마스크함수의 생성방법을 상기한 일반화된 과정에 따라 상기 도 6을 다시 참조하여 설명하면 다음과 같다. 실제로, 카자미 시퀀스는 서로 다른 특정의 m -시퀀스의 합으로 표현되어진 다는 것은 널리 알려진 사실이다. 따라서, 상기의 (32,7)코드를 생성하기 위해서는, 먼저, 길이 31인 골드 시퀀스를 생성하여야 하는데, 생성다항식 $x^5 + x^2 + 1$ 과 $x^5 + x^4 + x^2 + x + 1$ 로 생성되어지는 두 m -시퀀스의 합은 골드시퀀스가 된다. 또한, 각각의 m -시퀀스 $m(t)$ 는 생성다항식이 정해지면 <수학식5>과 같이 트레이스 함수(Trace function)를 이용하여 구할 수 있다.

<55> 【수학식 5】 $m_1(t) = \text{Tr}(A \alpha^t), t=0,1,\dots,30$

<56> 단,
$$\text{Tr}(a) = \sum_{n=0}^4 a^{2^n}, \quad a \in GF(2^5) \text{ 이다.}$$

<57> 상기 <수학식5>에서 A 는 m -시퀀스의 초기치에 따라 결정되는 값이다.

<58> 도 6은 상기의 시퀀스군중 골드 시퀀스군을 사용하여, (32,7)코드(즉, 7비트의 정보비트가 입력되면 32비트의 부호화심볼이 출력되는)를 생성하는 경우 상기와 같은 마스크함수의 생성과정을 나타낸다. 골드 시퀀스는 서로 다른 특정의 m-시퀀스의 합으로 표현되어진 다는 것은 널리 알려진 사실이다. 따라서, 상기의 (32,7)코드를 생성하기 위해서는, 먼저, 길이 31인 골드 시퀀스를 생성하여야 하는데, 생성다항식 $x^5 + x^2 + 1$ 과 $x^5 + x^4 + x^2 + x + 1$ 으로 생성되어지는 두 m-시퀀스의 합은 골드시퀀스가 된다.

<59> 도 6을 살펴보면, 610 단계에서 상기의 <수학식1>에 의해 생성다항식 $x^5 + x^2 + 1$ 로 생성되어지는 m-시퀀스 $m_1(t)$ 와 $x^5 + x^4 + x^2 + x + 1$ 으로 생성되어지는 m-시퀀스 $m_2(t)$ 를 구한다. 그러면, 520단계에서는 상기의 m-시퀀스 $m_2(t)$ 를 하기의 <수학식 6>에서 나타난 월시부호로 만드는 열치환 함수 $\sigma(t)$ 를 구한다.

<60> 【수학식 6】 $\sigma: \{0, 1, 2, \dots, 30\} \rightarrow \{1, 2, \dots, 31\}$

<61>
$$\sigma(t) = \sum_{i=0}^4 m_2(t) 2^{4-i}$$

<62> 그러면, 630단계에서 상기 m-시퀀스 $m_1(t)$ 를 0부터 30번까지 순회(cyclic shift)시켜 얻을 수 있는 31개의 시퀀스군을 상기의 $m_2(t)$ 를 월시부호로 만드는 열치환 함수 $\sigma(t)$ 의 역함수를 이용한 $\sigma^{-1}(t) + 2$ 로 열치환한 후, 각각의 시퀀스 맨 앞부분에 0을 덧붙임으로써 길이 32로 만들어서 길이 32인 31개의 시퀀스군 $d_i(t)$, $i=0, \dots, 31$, $t=1, \dots, 32$,을 생성한다. 상기와 같이 630단계에서 생성되어지는 시퀀스군은 <수학식 7>과 같이 수식으로 표현할 수 있다.

<63> 【수학식 7】 $\{d_i(t) | i=1, \dots, 32, t=0, \dots, 31\}$

$$\langle 64 \rangle \quad d_i(t) = \begin{pmatrix} 0, & \text{if } t=1 \\ m_d(t+i-2), & \text{if } t=1,3,\dots,32 \end{pmatrix}$$

$\langle 65 \rangle$ 상기 구해진 $d_i(t)$ 들은 상기 마스크 함수들로 31개의 마스크로 사용할 수 있다.

$\langle 66 \rangle$ 상기 구해진 $d_i(t)$ 들의 성질 중 한가지 성질은 상기의 마스크들중 두 개의 서로 다른 마스크들을 더하면 31개마스크 중 다른 하나의 마스크가 된다. 더 일반화시켜서 상기의 31개의 마스크들은 31개의 마스크중 특정한 5개의 마스크의 임의의 합으로 모두 표현되어질 수 있다.

$\langle 67 \rangle$ 상기 (32,7)코드를 생성할 때, 총 필요로하는 부호어(Code word)의 개수는 모든 경우의 입력 정보비트의 가지수인 $2^7 = 128$ 개이다. 이 때, 길이 32인 직교(orthogonal) 부호어의 개수는 32이고, 이 때, (32,7)코드를 생성하기 위해 필요한 마스크의 개수는 $(128/32)-1 = 3$ 개이다. 또한, 이 때, 3개의 마스크는 상기와 유사한 성질에 의해 2개의 마스크의 임의의 합으로 모두 표현되어질 수 있다. 따라서, 상기의 2개의 마스크를 고르는 방법이 필요하다. 상기의 2개의 마스크를 선택하는 방법을 설명하면, 상기 도 6의 630단계에서 $m_1(t)$ 를 0부터 31번까지 순회(cyclic shift)시켜 시퀀스군을 생성하는데, 이 때, $m_1(t)$ 를 i 번 순회(cyclic shift)시킨 m -시퀀스는 상기 <수학식1>을 사용하여 표현하면 $Tr(\alpha' \cdot \alpha')$ 가 된다. 즉, $m_1(t)$ 를 0부터 31번까지 순회(cyclic shift)시켜 시퀀스군은 초기치 A 가 $1, \alpha, \dots, \alpha^{30}$ 에 따라서 생성되어지는 시퀀스들이다. 이 때, 갈로아체의 원소 $1, \alpha, \dots, \alpha^{30}$ 중, 선형독립인 2개의 원소를 찾아서, 즉, 임의의 2개의 원소를 찾아서, 상기 2개의 원소를 초기치로 하는 시퀀스들을 선택하여 상기 도 6의 결과

값 들을 찾아 선택하면 상기와 같이 2개의 마스크의 임의의 합으로 3개 마스크 모두를 생성할 수 있다. 상기의 과정중 선형독립의 조건을 수학적식으로 나타내면 하기의 <수학식8>과 같다.

<68> 【수학식 8】 $\alpha, \beta, \gamma, \delta$: 선형 독립

<69> $\Leftrightarrow c_1\alpha + c_2\beta + c_3\gamma + c_4\delta \neq 0, \forall c_1, c_2, c_3, c_4$

<70> 실제로 갈로아체GF(2³)에서 1, alpha은 상기와 같은 2개의 선형독립인 원소로 널리 알려진 다항식 기저(polynomial basis)이다. 따라서, 상기 다항식 기저에 따라서 구해진 2개의 마스크 함수M1,M2는 다음과 같다.

<71> M1 = 0010 1000 0110 0011 1111 0000 0111 0111

<72> M2 = 0000 0001 1100 1101 0110 1101 1100 0111

<73> 하기 실시예에서는 상기에서 구해진 마스크를 이용한 부호기 및 복호기에 대해 설명할 것이다.

<74> 제 1 실시예

<75> 상기 제 1실시예에서는 상기의 생성방법에 따른 부호화 장치 및 방법을 제공한다. 도 7a는 상기 제1실시예에 따른 부호기를 나타낸 도면이다.

<76> 도 7a를 참조하면 10비트의 입력 정보비트들이 입력되면, 각각의 비트 a0,a1,a2,a3,a4,a5, a6들은 승산기 740,741,742,743,744,745,746에 입력된다. 이때, 월시부호 발생기 710에서는 길이 32인 월시부호 W1,W2,W4,W8,W16이 동시에 출력되는데, 월시부호 W1=0101 0101 0101 0101 0101 0101 0101 0101인 심볼들이 승산기 740에 입력되고, 한 심볼이 승산기 840으로 입력될 때마다, 승산기 740에

입력된 a0와 승산되어져 배타적 가산기 760로 출력된다. 월시부호 W2=0011 0011 0011 0011 0011 0011인 심볼들이 승산기 741에 입력되고, 한 심볼이 승산기 741으로 입력될 때마다, 승산기 741에 입력된 a1와 승산되어져 배타적 가산기 760로 출력된다. 월시부호 W4=00001111 00001111 00001111 00001111인 심볼들이 승산기 742에 입력되고, 한 심볼이 승산기 742으로 입력될 때마다, 승산기 742에 입력된 a2와 승산되어져 배타적 가산기 760로 출력된다. 월시부호 W8=00000000 11111111 00000000 11111111인 심볼들이 승산기 743 입력되고, 한 심볼이 승산기 743으로 입력될 때마다, 승산기 743에 입력된 a3와 승산되어져 배타적 가산기 760로 출력된다. 월시부호 W16= 0000000000000000 1111111111111111인 심볼들이 승산기 744에 입력되고, 한 심볼이 승산기 744으로 입력될 때마다, 승산기 744에 입력된 a4와 승산되어져 배타적 가산기 760로 출력된다.

<77> 또한, 마스크 생성기 720에서는 길이 32인 마스크 함수 M1,M2가 동시에 출력되는데, 마스크 함수 M1=0010 1000 0110 0011 1111 0000 0111 0111인 심볼들이 승산기 745에 입력되고, 한 심볼이 승산기 745으로 입력될 때마다, 승산기 745에 입력된 a7와 승산되어져 배타적 가산기 760로 출력된다. 마스크 함수 M2=0000 0001 1100 1101 0110 1101 1100 0111인 심볼들이 승산기 746에 입력되고, 한 심볼이 승산기 746으로 입력될 때마다, 승산기 746에 입력된 a8와 승산되어져 배타적 가산기 760로 출력된다. 여기서, 마스크 생성기 720은 주어진 규칙에 의해 마스크 함수를 생성할수도 있고, 미리 구해진 마스크 함수를 메모리에 저장해 둘수도 있다.

<78> 그러면, 배타적 가산기 760은 승산기 740,741,742,743,744,745,746으로부터 출력된 심볼들을 모두 배타적 가산하여 출력한다. 이때, 출력되어진 32개의 부호화 심볼들은 다시 심볼 천공기 770으로 입력되어 8심볼이 천공되어진 후 24심볼만이 출력되어진다. 상기 (24,7)부호기는 상기 (32,7) 부호기로부터 8심볼을 천공함으로써 얻어질 수 있다. 이 때, 상기 8개의 천공심볼 위치에 따라서, 상기 부호기의 최소거리가 달라지는데, 하기에 나타나는 8개의 천공위치는 모든 8개의 위치에 대한 조합 중에서도 우수한 성능을 나타내는 위치를 나타내고, 하기 표 1에 나타나는 천공위치를 사용하면 최소거리가 9를 가지게 되고, 웨이트 분포도 우수하다.

<79> 【표 1】

0, 1, 3, 7,10,13,15,18
0, 1,5 3, 7,10,13,15,25
0, 4,13,18,20,25,29,31
0, 1, 3, 7,10,11,13,16
0, 3,10,11,13,15,26,31

<80> 도 8은 상기 부호기의 동작에 대한 플로우 차트를 나타낸다. 도 8을 참조하면, 먼저 800단계에서 7비트의 정보비트열 a_0, a_1, \dots, a_6 를 입력하고, 부호기에서 최종적으로 출력할 부호심볼을 나타낼 32개의 변수인 $code[]$ 를 0으로 초기화 하고, 변수 i 를 0으로 초기화한다.

<81> 그러면, 810단계에서는 첫 번째 정보비트 a_0 이 1일 때, 일시부호 $W1 = 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101$ 를 길이 32인 부호심볼열 변수 $code[]$ 에 배타적 가산해주는 과정이다. 이 과정이 끝나면, 812단계에서는 두 번째 정보비트 a_1 가 1일 때, 일시부호 $W2=0011\ 0011\ 0011\ 0011\ 0011\ 0011\ 0011\ 0011$ 를 길이 32

인 부호심볼열 변수 `code[]`에 배타적 가산해주는 과정을 수행한다. 이 과정이 끝나면, 814단계에서는 세 번째 정보비트 a_2 가 1일 때, 월시부호 $W_4=00001111$ 00001111 00001111 00001111 00001111 00001111 00001111를 길이 32인 부호심볼열 변수 `code[]`에 배타적 가산해주는 과정을 수행한다. 이 과정이 끝나면, 816단계에서는 네 번째 정보비트 a_4 가 1일 때, 월시부호 $W_8=0000000011111111$ 0000000011111111를 길이 32인 부호심볼열 변수 `code[]`에 배타적 가산해주는 과정을 수행한다. 이 과정이 끝나면, 818단계에서는 다섯 번째 정보비트 a_4 가 1일 때, 월시부호 $W_{16}=0000000000000000$ 1111111111111111를 길이 32인 부호심볼열 변수 `code[]`에 배타적 가산해주는 과정을 수행한다. 그러면, 820단계에서는 여섯 번째 정보비트 a_5 가 1일 때, 마스크 함수 $M_1 = 0010\ 1000\ 0110\ 0011\ 1111\ 0000\ 0111\ 0111$ 를 길이 32인 부호심볼열 변수 `code[]`에 배타적 가산해주는 과정을 수행하고, 822단계에서는 일곱 번째 정보비트 a_6 이 1일 때, 마스크 함수 $M_2 = 0000\ 0001\ 1100\ 1101\ 0110\ 1101\ 1100\ 0111$ 를 길이 32인 부호심볼열 변수 `code[]`에 배타적 가산해주는 과정을 수행한다. 상기의 과정이 끝나면 상기의 7비트의 정보비트에 대응되는 7개의 길이 32인 수열 $W_1, W_2, W_4, W_8, W_{16}, M_1, M_2$ 들중 1인 정보비트들에 대응되는 수열들만을 모드 배타적가산한 부호화 심볼 변수의 값 `code[]`이 생성된다.

<82> 도 9는 상기 도 7a 및 도 7b의 부호기에 대응하는 복호기를 도시하고 있다. 도 9을 살펴보면, 먼저 수신신호 $r(t)$ 가 입력되면, 상기 수신신호 $r(t)$ 는 0 삽입기 950에 입력되어져 송신단 부호기의 천공기 770에서 천공한 심볼위치에 0를 삽입하여 32개의 심볼로 구성되어진 수신신호로 출력되어진다. 그러면, 상기 0

삽입기950에서 출력되어진 길이 32인 수신신호는 3개의 승산기 901,902,903와 상관도 계산기920에 입력된다. 그러면, 마스크 생성기910은 3개의 모든 경우의 마스크를 생성하여 승산기901,902,903에 출력한다. 여기서, 상기 마스크 생성기 910은 마스크 함수를 주어진 규칙에 의해 생성할수도 있고, 미리 메모리에 저장해둘 수도 있다. 이 때, 승산기 901은 입력된 수신신호 $r(t)$ 와 마스크 생성기 910으로부터 출력된 마스크함수 $M1$ 을 승산하여 상관도 계산기 921에 출력하고, 승산기 902는 입력된 수신신호 $r(t)$ 와 마스크 생성기 910으로부터 출력된 마스크함수 $M2$ 를 승산하여 상관도 계산기942에 출력하고, 승산기 903은 입력된 수신신호 $r(t)$ 와 마스크 생성기 910으로부터 출력된 마스크함수 $M3$ 를 승산하여 상관도 계산기 923에 출력한다. 즉, 수신신호 자신과 가능한 3개의 모든 마스크함수들이 수신신호 $r(t)$ 와 승산되어진, 총 4가지의 신호들이 4개의 상관도 계산기들 920,921,922,923에 입력된다. 그러면, 상기 상관도 계산기920은 입력된 수신신호 $r(t)$ 를 32개의 모든 길이 32인 월시부호들과의 32가지 상관도를 구하여(다시 말해 역 하다마드 방식을 통해 상관도를 구함) 상관도의 값이 가장 큰 값과, 그 때의 월시부호 인덱스, 그리고, 상관도 계산기의 인덱스 0을 상관도 비교기940에 출력한다. 이 때, 상기 상관도 계산기의 인덱스는 상기 상관도 계산기에 입력된 신호가 수신신호에 몇 번째 마스크 함수가 승산되어져 입력되었는지를 나타내는 마스크 함수의 인덱스와 동일하다. 그러나, 마스크 인덱스가 0이라함은 아무런 마스크도 곱해지지 않았음을 의미한다. 그리고, 이와 동시에 상관도 계산기921은 입력된 수신신호 $r(t)$ 와 마스크 함수 $M1$ 을 승산한 신호를 32개의 모든 길이 32인 월시부호와의 32가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산

되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 1을 상관도 비교기940에 출력한다. 그리고, 상관도 계산기922는 입력된 수신신호 $r(t)$ 와 마스크 함수 M2를 승산한 신호를 32개의 모든 길이 32인 월시부호와의 32가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 2를 상관도 비교기940에 출력한다. 상관도 계산기923은 입력된 수신신호 $r(t)$ 와 마스크 함수 M3을 승산한 신호를 32개의 모든 길이 32인 월시부호와의 32가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 3을 상관도 비교기940에 출력한다.

<83> 그러면, 상기 상관도비교기940은 상기 상관도계산기들920,921,922,923에서 입력된 4가지의 최대 상관값을 비교하여 가장 최대인 상관값을 구하고, 그 상관값에 대해 해당 상관도 계산기로부터 입력되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스, 즉, 수신신호 $r(t)$ 와 승산되어진 마스크함수의 인덱스를 가지고, 수신신호에 대한 복호신호를 결정하여 출력한다.

<84> 도 10은 상기의 상관도비교기940의 동작을 플로우 차트로 도시하고 있다.

<85> 도 10을 참조하면, 먼저 1000단계에서는 회수를 나타내는 인덱스 I은 1로, 검색하려는 최대값, 월시부호 인덱스, 마스크인덱스들은 0으로 초기화 한다. 그러면, 1010단계에서는 1번째 상관도 계산기920으로부터 입력된 상관도의 1번째 최대값, 월시부호 인덱스, 마스크 인덱스들을 입력한다. 그러면, 1020단계에서 상기 1번째 최대값과 상기 최대값을 비교하여 1번째 최대값이 크면 1030단계로 진행하여 최대값에 1번째 최대값을, 월시부호 인덱스에는 1번째 상관도 계산기로

부터 입력된 월시부호 인덱스를, 마스크 인덱스에는 1번째 상관도 계산기로부터 입력된 마스크 인덱스를 차례로 저장하고, 1040단계를 진행하고, 1020에서 상기 1번째 최대값과 최대값을 비교하여 1번째 최대값이 작으면 1040단계를 바로 진행한다. 이 때, 1040단계에서는 회수를 나타내는 인덱스 i 가 상기 상관도계산기의 개수인 4인지를 판단한다. 1040단계에서는 회수를 나타내는 인덱스 i 가 상기 상관도계산기의 개수인 4가 아니면 1060단계를 진행하여 회수를 나타내는 인덱스 i 를 1씩 증가시키고, 다시 1010단계를 진행하여 2번째 상관도 계산기921로부터 상관도의 2번째 최대값, 월시부호 인덱스, 마스크 인덱스들을 입력하고 상기와 같은 과정을 반복한다. 상기와 같이 반복하다가 4번째 상관도 계산기로부터 입력된 상관값이 모두 비교되면 이 때의 회수를 나타내는 i 가 4일 것이므로, 1040단계에서 i 가 4인지를 판단하여 통과하면 1050단계로 진행하여 상기 변수인 월시부호 인덱스와 마스크 인덱스에 대응하는 복호비트들(7비트)을 출력한다.

<86> 상기한 제1실시 예에서는 32개의 부호심볼을 발생하고 8개의 심볼을 천공하여 24심볼을 생성하는 부호기를 설명하고 있다. 한편, 후술될 제2실시예에서는 상기 도 7a와 달리 월시부호생성기710 및 마스크 생성기720에서 미리 8심볼을 천공하여 24심볼을 출력하는 부호화기에 대해 설명한다.

<87> 제2실시예

<88> 상기 제 2실시예에서는 월시부호생성기710 및 마스크 생성기720에서 미리 8심볼을 천공하여 24심볼을 출력하는 부호화 장치 및 방법을 제안한다. 상기의 부호화 장치의 구조는 제1실시예의 부호화구조와 유사하다. 단, 월시부호 발생기와

마스크 생성기에서 출력되는 수열들은 상기 제1 실시예에서의 수열들의

0,1,3,7,10,13,15,18번째의 항을 천공함으로써 길이 24을 가진다.

<89> 상기 도 7b을 참조하면, 7비트의 입력 정보비트들이 입력되면, 각각의 비트 $a_0, a_1, a_2, a_3, a_4, a_5, a_6$ 들은 승산기 740, 741, 742, 743, 744, 745, 746에 입력된다. 이 때, 월시부호 발생기 710에서는 길이 32인 천공된 월시부호 $W_1, W_2, W_4, W_8, W_{16}$ 이 동시에 출력되는데, 천공된 월시부호 $W_1 = 0010\ 0110\ 0011\ 0101\ 0101\ 0101$ 인 심볼들이 승산기 740에 입력되고, 한 심볼이 승산기 740으로 입력될 때마다, 승산기 740에 입력된 a_0 와 승산되어져 배타적 가산기 760로 출력된다. 천공된 월시부호 $W_2 = 1001\ 0010\ 1001\ 0011\ 0011\ 0011$ 인 심볼들이 승산기 741에 입력되고, 한 심볼이 승산기 741로 입력될 때마다, 승산기 741에 입력된 a_1 와 승산되어져 배타적 가산기 760로 출력된다. 천공된 월시부호 $W_4 = 0111\ 0001\ 1000\ 1111\ 0000\ 1111$ 인 심볼들이 승산기 742로 입력되고, 한 심볼이 승산기 742로 입력될 때마다, 승산기 742에 입력된 a_2 와 승산되어져 배타적 가산기 760로 출력된다. 천공된 월시부호 $W_8 = 0000\ 1111\ 1000\ 0000\ 1111\ 1111$ 인 심볼들이 승산기 743로 입력되고, 한 심볼이 승산기 743으로 입력될 때마다, 승산기 743에 입력된 a_3 와 승산되어져 배타적 가산기 760로 출력된다. 천공된 월시부호 $W_{16} = 0000\ 0000\ 0111\ 1111\ 1111\ 1111$ 인 심볼들이 승산기 744로 입력되고, 한 심볼이 승산기 744로 입력될 때마다, 승산기 744에 입력된 a_4 와 승산되어져 배타적 가산기 760로 출력된다.

<90> 마스크 생성기 720에서는 길이 24인 천공된 마스크 함수 M_1, M_2 이 동시에 출력되는데, 천공된 마스크 함수 $M_1 = 1100\ 0100\ 1111\ 0000\ 0111\ 0111$ 인 심볼들이 승

산기 745로 입력되고, 한 심볼이 승산기 745로 입력될 때마다, 승산기 745에 입력된 a5와 승산되어져 배타적 가산기 760로 출력된다. 천공된 마스크 함수 $M2=0000\ 1101\ 0010\ 1101\ 1100\ 0111$ 인 심볼들이 승산기 746에 입력되고, 한 심볼이 승산기 746으로 입력될 때마다, 승산기 746에 입력된 a6와 승산되어져 배타적 가산기 760로 출력된다. 그러면, 배타적 가산기 760은 승산기 740, 741, 742, 743, 744, 745, 746으로부터 출력된 심볼들을 모두 가산하여 출력한다.

<91> 도 11은 상기 부호기의 동작에 대한 플로우 차트를 도시하고 있다. 도 11을 참조하면, 먼저 1100단계에서 7비트의 정보비트열 a_0, a_1, \dots, a_6 를 입력하고, 부호기에서 최종적으로 출력할 부호심볼을 나타낼 24개의 변수인 $code[]$ 를 0으로 초기화 하고, 변수 i 를 0으로 초기화한다. 그러면, 1110단계에서는 첫 번째 정보비트 a_0 이 1일 때, 천공된 월시부호 $W1 = 0010\ 0110\ 0011\ 0101\ 0101\ 0101$ 를 길이 24인 부호심볼열 변수 $code[]$ 에 배타적 가산해주는 과정을 수행한다. 이 과정이 끝나면, 1112단계에서는 두 번째 정보비트 a_1 가 1일 때, 월시부호 $W2 = 1001\ 0010\ 1001\ 0011\ 0011\ 0011$ 를 길이 24인 부호심볼열 변수 $code[]$ 에 배타적 가산해주는 과정을 수행한다. 이 과정이 끝나면, 1114단계에서는 세 번째 정보비트 a_2 가 1일 때, 월시부호 $W4 = 0111\ 0001\ 1000\ 1111\ 0000\ 1111$ 를 길이 24인 부호심볼열 변수 $code[]$ 에 배타적 가산해주는 과정을 수행한다. 이 과정이 끝나면, 1116단계에서는 네 번째 정보비트 a_3 이 1일 때, 월시부호 $W8 = 0000\ 1111\ 1000\ 0000\ 1111\ 1111$ 를 길이 24인 부호심볼열 변수 $code[]$ 에 배타적 가산해주는 과정을 수행한다. 이 과정이 끝나면, 1118단계에서는 다섯 번째 정보비트 a_4 가 1일 때, 월

시부호 W16 = 0000 0000 0111 1111 1111 1111를 길이 24인 부호심볼열 변수 code[]에 배타적 가산해주는 과정을 수행한다. 그러면, 1120단계에서는 여섯번째 정보비트 a5가 1일 때, 마스크 함수 M1 = 1100 0100 1111 0000 0111 0111를 길이 24인 부호심볼열 변수 code[]에 배타적 가산해주는 과정을 수행하고, 1122단계에서는 일곱번째 정보비트 a6이 1일 때, 마스크 함수 M2 = 0000 1101 0010 1101 1100 0111를 길이 24인 부호 심볼열 변수 code[]에 배타적 가산해주는 과정을 수행한다. 상기의 과정이 끝나면 상기의 10비트의 정보비트에 대응되는 10개의 길이 32인 수열 W1, W2, W4, W8, W16, M1, M2들중 1인 정보비트들에 대응되는 수열들만을 모드 배타적가산한 부호화 심볼 변수의 값 code[]이 생성된다.

<92> 상기 (24,7)부호기는 상기 제1실시에 (24,7)부호기의 모든 부호어들에 대해서 0,1,3,7,10,13,15,18번째 심볼들을 천공하여 만든 128개의 부호어로 구성되어 있다. 상기 24개의 심볼들을 가지고 복호하는 복호기를 상기 도 9를 참조하여 설명하면 다음과 같다. 여기서, 특히 상기 복호기는 24개의 심볼들을 복호할 수 있도록 구성되기 때문에, 상술한 제1실시 예에서처럼 0삽입기 950이 필요하지 않다.

<93> 상기 도 9를 참조하면, 먼저 수신신호 r(t)가 입력되면 상기 수신신호 r(t)는 7개의 승산기 901, 902, 903와 상관도 계산기 920에 입력된다. 그러면, 마스크 생성기 910은 3개의 모든 경우의 마스크를 생성하여 승산기 901, 902, 903에 출력한다. 이 때, 승산기 901은 입력된 수신신호 r(t)와 마스크 생성기 910으로부터 출력된 마스크 함수 M1을 승산하여 상관도 계산기 921에 출력한다. 승산기 902는 입력된 수신신호 r(t)와 마스크 생성기 910으로부터 출력된 마스크 함수 M2를 승산하

여 상관도 계산기2 942에 출력한다. 승산기 903은 입력된 수신신호 $r(t)$ 와 마스크 생성기 910으로부터 출력된 마스크함수 $M3$ 을 승산하여 상관도 계산기7 923에 출력하면, 수신신호 자신과 가능한 3개의 모든 마스크함수들이 수신신호 $r(t)$ 와 승산되어진, 총 4가지의 신호들이 4개의 상관도 계산기0-3에 입력된다. 그러면, 상관도 계산기0은 입력된 수신신호 $r(t)$ 를 32개의 모든 길이 24인 월시부호와의 32가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 0을 상관도 비교기940에 출력한다. 이 때, 상기 상관도 계산기의 인덱스는 상기 상관도 계산기에 입력된 신호가 수신신호에 몇번째 마스크 함수가 승산되어져 입력되었는지를 나타내는 마스크 함수의 인덱스와 동일하다. 그러나, 마스크 인덱스가 0이라함은 아무런 마스크도 곱해지지 않았음을 의미한다. 그리고, 이와 동시에 상관도 계산기1은 입력된 수신신호 $r(t)$ 와 마스크 함수 $M1$ 을 승산한 신호를 32개의 모든 길이 24인 월시부호와의 32가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 1을 상관도 비교기 940에 출력한다. 상관도 계산기2는 입력된 수신신호 $r(t)$ 와 마스크 함수 $M2$ 를 승산한 신호를 32개의 모든 길이 24인 월시부호와의 32가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 2를 상관도 비교기940에 출력한다. 상관도 계산기3는 입력된 수신신호 $r(t)$ 와 마스크 함수 $M3$ 을 승산한 신호를 32개의 모든 길이 24인 월시부호와의 32가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진

월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 3을 상관도 비교기940에 출력한다.

<94> 그러면, 상관도비교기940은 상기 상관도 비교기0-3에서 입력된 4가지의 최대

상관값을 비교하여 가장 최대인 상관값을 구하면, 그 상관값에 대해 해당 상관도 계산기로부터 입력되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스, 즉, 수신신호 $r(t)$ 와 송신되어진 마스크함수의 인덱스를 가지고, 수신신호에 대한 복호신호를 결정하여 출력한다. 도 10은 상기의 비교기990의 동작을 플로우 차트로 나타낸 것이다. 도 10을 참조하면, 먼저 1000단계에서는 회수를 나타내는 인덱스 I 은 1로, 검색하려는 최대값, 월시부호 인덱스, 마스크인덱스들은 0으로 초기화한다. 그러면, 1010단계에서는 1번째 상관도 계산기920으로부터 출력된 상관도의 1번째 최대값, 월시부호 인덱스, 마스크 인덱스들을 입력한다. 그러면, 1020에서 상기 1번째 최대값과 상기 최대값을 비교하여 1번째 최대값이 크면 1030단계로 진행하여 최대값에 1번째 최대값을, 월시부호 인덱스에는 1번째 상관도 계산기로부터 입력된 월시부호 인덱스를, 마스크 인덱스에는 1번째 상관도 계산기로부터 입력된 마스크 인덱스를 차례로 저장한후 1040단계를 진행하고, 상기 1020단계에서 상기 1번째 최대값과 최대값을 비교하여 1번째 최대값이 작으면 1040단계를 바로 진행한다. 이 때, 1040단계에서는 회수를 나타내는 인덱스 i 가 상기 상관도계산기의 개수인 4인지를 판단한다. 1040단계에서는 회수를 나타내는 인덱스 i 가 상기 상관도계산기의 개수인 4가 아니면 1060단계를 진행하여 회수를 나타내는 인덱스 i 를 1씩 증가시키고, 다시 1010단계를 진행하여 2번째 상관도 계산기에서 입력된 922에서 출력된 상관도의 2번째 최대값, 월시부호 인덱스, 마스크 인덱스들을 입력하고 상기와 같은 과정을 반복한다. 상기와 같이 반복하다가 4번째 상관도 계산기로부터 입력된 상관값이 모두 비교되면 이 때의 회수를 나타내는 i 가 4일 것이므로 1040단계에서 i 가

4인지를 판단하여 통과하면 1050단계로 진행하여 상기 변수인 월시부호 인덱스와 마스크 인덱스에 대응하는 복호비트들을 출력한다.

<95> 한편 본 발명의 상세한 설명에서는 구체적인 실시 예에 관해 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로 본 발명의 범위는 설명된 실시 예에 국한되어 정해져서는 안되며 후술하는 특허청구의 범위뿐만 아니라 이 특허청구의 범위와 균등한 것들에 의해 정해져야 한다.

【발명의 효과】

<96> 상술한 바와 같이, 본 발명은 퀄리티 메칭 지시자(Quality Matching Indicator) 비트를 사용하는 부호분할다중접속 통신시스템에서 퀄리티 메칭 지시자(Quality Matching Indicator)를 오류정정방식의 블록부호를 이용해 최적 부호화할수 있는 이점을 제공한다.

【특허청구범위】**【청구항 1】**

k개의 정보비트의 입력으로 n개의 부호심볼을 출력하는 블록 부호 생성장치에 있어서,

상기 k개의 정보비트 입력으로 상기 n개보다 많은 부호심볼을 출력하는 확장된 리드물러(Extended Reed Muller) 부호기와,

상기 부호기로부터 출력되는 상기 부호심볼들 중 미리 주어지는 천공패턴에 의해 소정의 개수를 천공하여 상기 n개의 부호심볼을 출력하는 천공기를 포함하는 것을 특징으로 하는 장치.

【청구항 2】

제1항에 있어서, 상기 확장된 리드물러 부호기는,

골드시퀀스를 소정 규칙에 의해 열치환하여 얻어진 특정 시퀀스를 발생시키는 마스크 발생기와,

상기 골드시퀀스를 소정 규칙에 의해 열치환하여 얻어진 m-시퀀스를 발생시키는 월시부호 발생기와,

상기 마스크 발생기의 출력과 상기 월시부호 발생기의 출력을 배타적 가산하는 가산기를 포함하는 것을 특징으로 하는 장치.

【청구항 3】

제2항에 있어서,
상기 블록 부호 생성장치는 (24,7) 블록 부호기인 것을 특징으로 하는
장치.

【청구항 4】

제3항에 있어서, 상기 특정 시퀀스는,
M1 = 0010 1000 0110 0011 1111 0000 0111 0111,
M2 = 0000 0001 1100 1101 0110 1101 1100 0111,
및 상기 M1와 M2를 배타적 가산한 시퀀스 중 어느 하나인 것을 특징으로 하
는 장치.

【청구항 5】

제4항에 있어서, 상기 미리 주어지는 천공패턴은,
0, 1, 3, 7,10,13,15,18,
0, 1,5 3, 7,10,13,15,25,
0, 4,13,18,20,25,29,31,
0, 1, 3, 7,10,11,13,16,
0, 3,10,11,13,15,26,31 중 어느 하나인 것을 특징으로 하는 장치.

【청구항 6】

제1항에 있어서, 상기 확장된 리드물리 부호기는,

월시부호들중 기저(basis)가 되는 소정의 월시부호들을 발생하는 월시부호 생성기와,

골드시퀀스를 소정 규칙에 의해 열치환하여 얻어진 소정의 마스크 신호들을 발생하는 마스크 생성기와,

상기 입력되는 k개의 정보비트들과 상기 월시부호 생성기 및 상기 마스크 생성기로부터의 출력을 일대일로 승산하는 복수의 승산기들과,

상기 승산기들의 출력을 배타적 가산하여 출력하는 가산기를 포함하는 것을 특징으로 하는 장치.

【청구항 7】

고속 패킷 데이터 전송을 지원하는 부호분할다중접속 통신시스템에서 퀄리티 매칭 지시자(QMI)를 부호화하기 위한 장치에 있어서,

7 비트로 표현되는 퀄리티 매칭 지시자를 입력하여 블록 부호화하여 32개의 부호심볼을 생성하는 블록 부호기와,

상기 부호기로부터의 상기 32개의 부호심볼들중 8개를 천공하여 24개의 부호심볼을 출력하는 천공기를 포함하는 것을 특징으로 하는 장치.

【청구항 8】

제7항에 있어서, 상기 확장된 리드물리 부호과정은,

골드시퀀스를 소정 규칙에 의해 열치환하여 얻어진 특정 시퀀스를 발생하는 마스크 발생과정과,

상기 골드시퀀스를 소정 규칙에 의해 열치환하여 얻어진 m-시퀀스를 발생하는 월시부호 발생과정과,

상기 특정 시퀀스와 상기 m시퀀스를 배타적 가산하여 상기 32개의 부호심볼을 생성하는 과정을 포함하는 것을 특징으로 하는 방법.

【청구항 9】

제8항에 있어서, 상기 특정 시퀀스는,

M1 = 0010 1000 0110 0011 1111 0000 0111 0111,

M2 = 0000 0001 1100 1101 0110 1101 1100 0111,

및 상기 M1와 M2를 배타적 가산한 시퀀스 중 어느 하나인 것을 특징으로 하는 방법.

【청구항 10】

제9항에 있어서, 상기 미리 주어지는 천공패턴은,

0, 1, 3, 7,10,13,15,18,

0, 1,5 3, 7,10,13,15,25,

0, 4, 13, 18, 20, 25, 29, 31,

0, 1, 3, 7, 10, 11, 13, 16,

0, 3, 10, 11, 13, 15, 26, 31 중 어느 하나인 것을 특징으로 하는 방법.

【청구항 11】

월시부호들 및 상기 월시부호들과 소정 마스크함수들의 합으로 만들어지는 부호들로 구성되는 부호를 복호하기 위한 복호장치에 있어서,

소정 마스크 신호들을 출력하는 마스크 생성기와,

수신신호와 모든 경우 월시부호들을 상관하고, 상관도의 값이 가장 큰 값, 그 때의 월시부호의 인덱스 및 상관도 계산기의 인덱스를 출력하는 제1 상관도 계산기와,

수신신호와 상기 마스크 생성기로부터의 상기 마스크 신호들을 일대일 승산하는 승산기들과,

상기 승산기들중 대응되는 승산기의 출력을 모든 경우의 월시부호들과 상관하고, 상관도의 값이 가장 큰 값, 그 때의 월시부호의 인덱스 및 상관도 계산기의 인덱스를 출력하는 복수의 상관도 계산기들과,

상기 제1 상관도 계산기 및 상기 복수의 상관도 계산기들의 출력을 입력하고, 상관값들을 비교하여 가장 큰 상관값에 대응하는 월시부호 인덱스, 상관도

계산기의 인덱스를 가지고 수신신호의 복호신호를 결정하여 출력하는 상관도비교기를 포함하는 것을 특징으로 하는 장치.

【청구항 12】

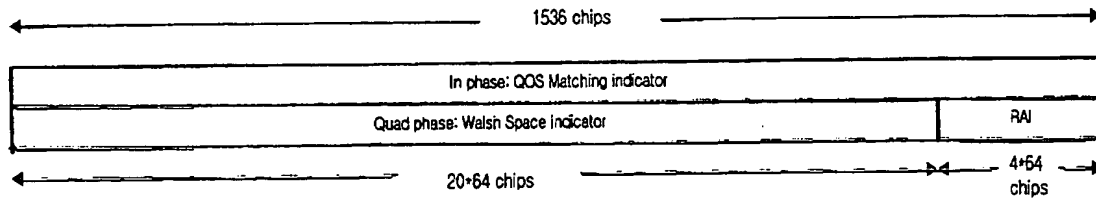
k개의 정보비트를 가지고 n개의 부호심볼을 생성하는 블록 부호 생성방법에 있어서,

상기 k개의 정보비트들을 가지고 상기 n개보다 많은 부호심볼을 생성하는 확장된 리드물러(Extended Reed Muller) 부호 과정과,

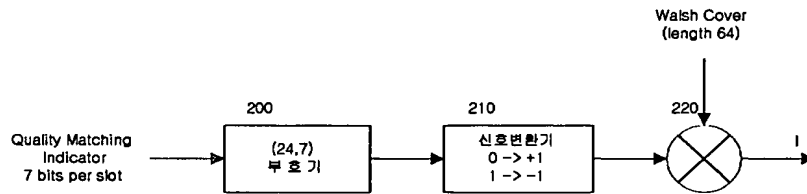
상기 생성된 부호심볼들 중 미리 주어지는 천공패턴에 의해 소정의 개수를 천공하여 상기 n개의 부호심볼을 생성하는 과정을 포함하는 것을 특징으로 하는 방법.

【도면】

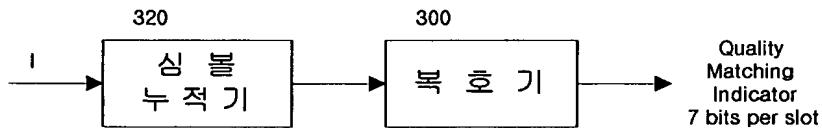
【도 1】



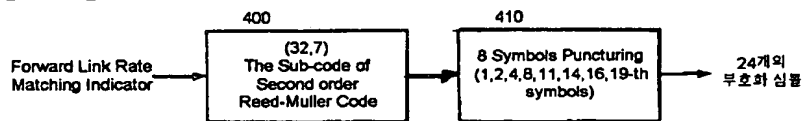
【도 2】



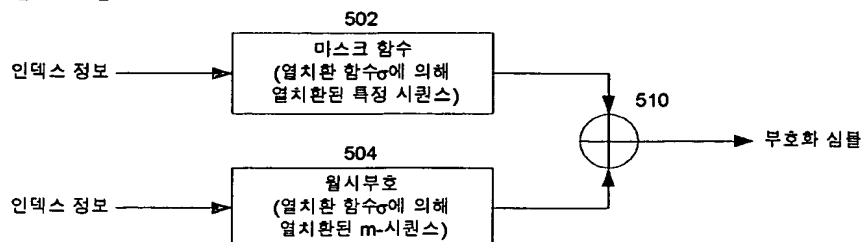
【도 3】



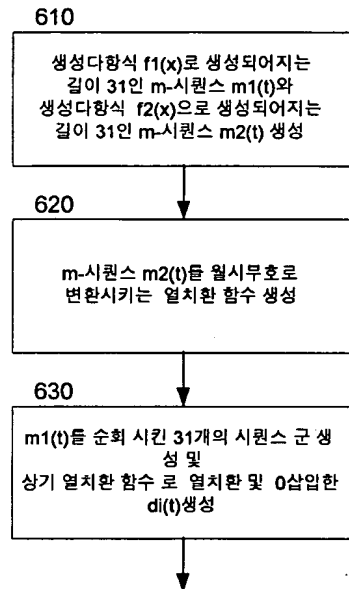
【도 4】



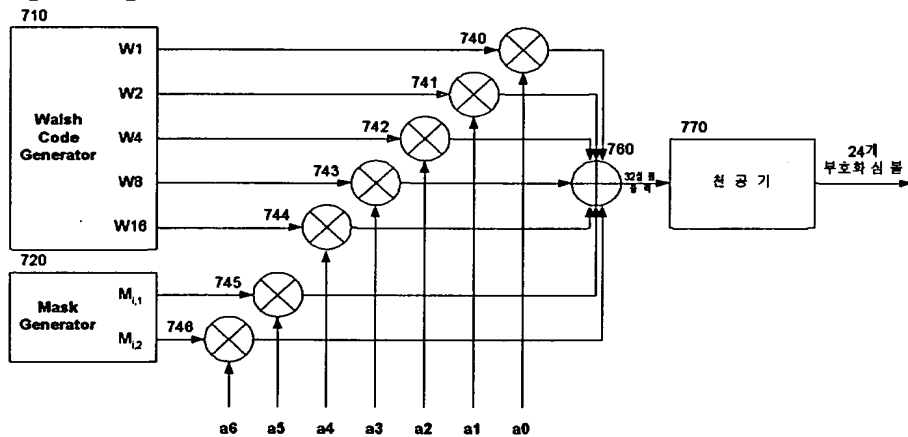
【도 5】



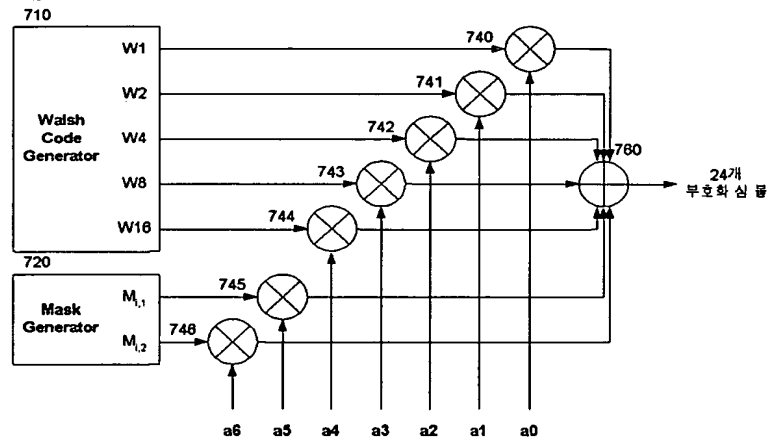
【도 6】



【도 7a】



【도 7b】

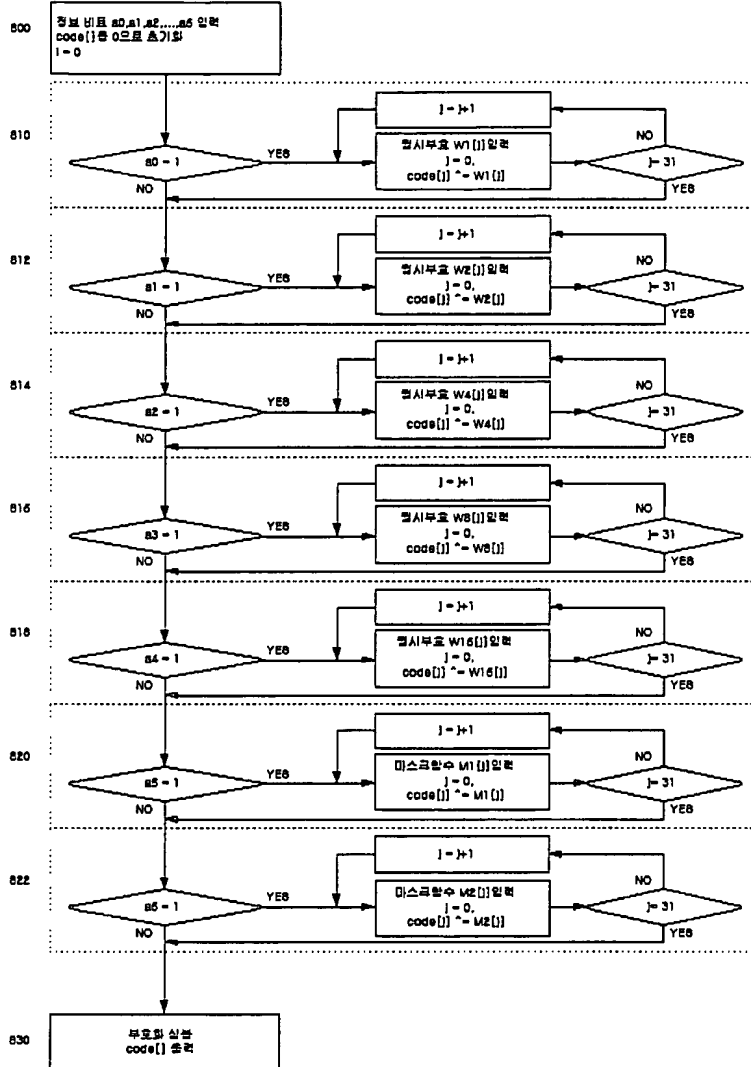




1020000062084

출력 일자: 2001/10/26

【도 8】

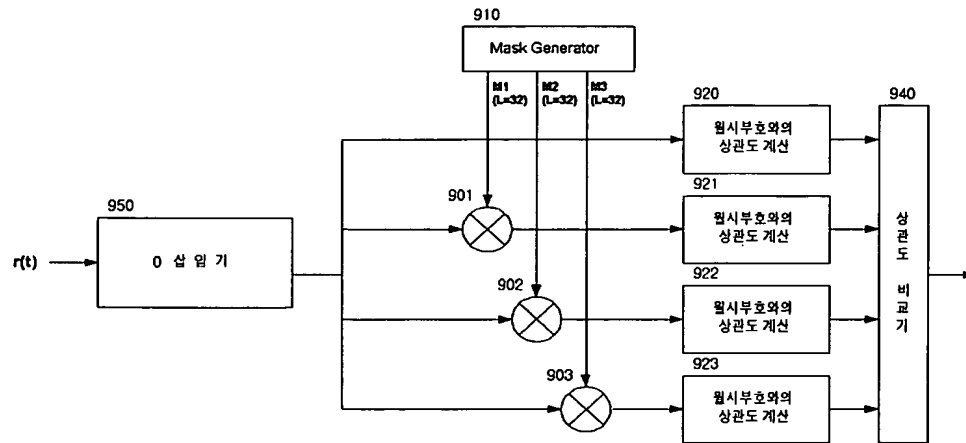




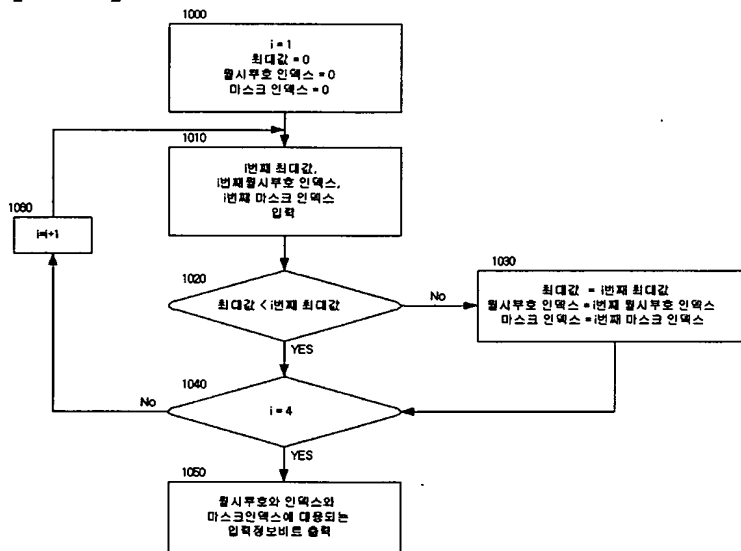
1020000062084

출력 일자: 2001/10/26

【도 9】



【도 10】





1020000062084

출력 일자: 2001/10/26

【도 11】

